Mobile Applications Security (3 hours)

Lila Ghemri

Department of Computer Science

Texas Southern University

ghemri_lx@tsu.edu

This work was supported by NSF grants 1241772

Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not reflect the views of the National Science Foundation

Objectives

- Understand the security and privacy threats to mobile applications

- Understand the basic strategies and approaches to enhance mobile application security and privacy.

Lesson Plan

- Topics
    - Coding Vulnerability
    - Java Mobile Security
    - WAP and Mobile HTML Security
    - Mobile Geolocation

Vulnerabilities

- A vulnerability is a security hole in a piece of software, hardware or operating system that provides a potential angle to attack the system.
- A vulnerability can be as simple as weak passwords or as complex as buffer overflows or SQL injection vulnerabilities.

Exploit and Payload

What is an exploit?

- A small and highly specialized computer program who has been designed to take advantage of a specific vulnerability and to provide access to a computer system.
- Exploits often deliver a payload to the target system to grant the attacker access to the system.

What is a payload?

- A payload is the piece of software that let an attacker control a computer system after it has been exploited. The payload is typically attached to and delivered by the exploit.

Discovering vulnerabilities

Penetration testing:

- Penetration testing, often called “pentesting”,“pen testing”, or “security testing”, is the practice of attacking your own IT system in the same way a hacker would to identify security holes.
- The difference between Hacking and Pentesting is that the latter is done with the system’s owner permission.

Remote Code Execution

- This vulnerability allows rogue code to be run on an “unprotected” server and retrieve information contained in that server.
- Improper coding errors lead to this vulnerability.
- Some programming languages are more prone to vulnerabilities than others.

Remote Code vulnerability

Global variables: Global variables have access to every segment of a program.

Register variables are system variables that have access to the CPU.

Defining a Register Global variable allows the user to initialize variables remotely.

Non-initialized parameters can be used to include unwanted files from an attacker and trigger their execution remotely.

The flaw comes from passing unsanitized user input to a function that would then be executed by the Web app.

Mitigations

- Limit the use of register_global variables.
- If using, make sure that it is properly initialized.
- It is an absolute must to sanitize user input before processing it.
- Avoid using shell commands.

SQL Injections

- SQL injections are common and popular vulnerabilities.
- This technique allows an attacker to retrieve information from a Web server’s database.
- This is done through substituting malicious code for variable values in SQL code.

Mitigations for SQL injections

- Avoid connecting to that database as a super-user or as the owner.
- Sanitize user input.
- Limit the use of open variables that can be initialized by client code.

Format String Vulnerabilities

- This issue results from the use of the string datatype for user input.
- A malicious user may use generic format tokens, to print data from the stack or possible other locations in memory.
- Problems: Denial-of-service caused by reading attacks to read memory until an illegal address causes the program to crash.
- Writing attacks to the execution pointer and force execution of malicious code.

Mitigation

- Check user input
- Put the correct formatting specifiers in code.
- Check for size and type of data versus its declared datatype.

Java Mobile Security

The Java Mobile Edition (JME) is one of the most popular development platforms for Mobile Applications.

- Good Security history, inherited from the Java platform
- Security Approach comprises 3 main principles:
  - Sandbox applications and prevent them from interacting with each other
  - Limit application’s hardware access
  - Ask the user.

Configurations and Profiles

- JME is highly configurable to work for a specific phone and a specific carrier.
- Configuration standards define the minimum capabilities (speed and memory)
- Profiles extend capabilities and add functionality for a specific use.

Enabling Security in JVM

- The JME security specifies groups access to a device using permission domains.
- No app should run using the maximum permission.
- Always design your app with the minimal permission in mind and only increase group access if needed.
- Native functionality –code that executes outside of JVM is prohibited.
- All classes must come from the same JAR file to prevent cross app loading
- Signatures are used to verify the integrity and origin of an application and decide how much to trust it.

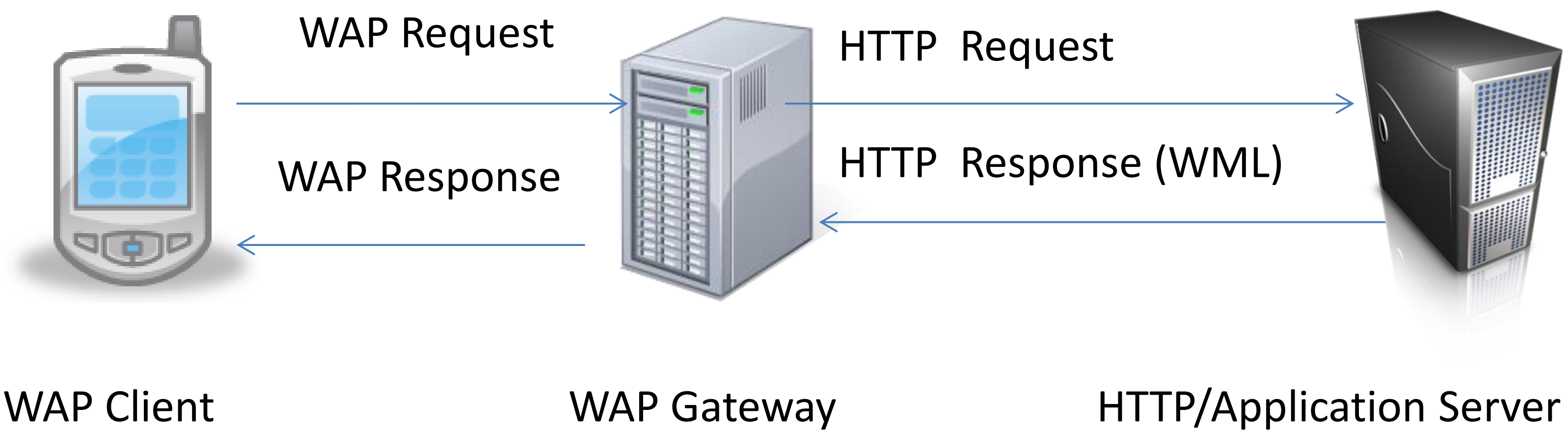
WAP and Mobile HTML Security

- Wireless Application Protocol(WAP): Provides a method to access the Internet from mobile devices.
- Based on Wireless Markup Language. Usually targeted for mobile devices that are not smartphones.
- WAP 2.0 supports xHTML, CSS and Transport Layer Security (TLS).
- Mobile HTML: Slimmed down versions of regular websites for mobile use. Mobile HTML sites are growing in popularity as more devices have an increase in their capabilities.

WAP Architecture

WAP architecture includes:

- WAP Browser for the mobile device
- Destination application HTTP Web server
- WAP gateway : acts like a proxy server between a mobile device and the HTTP server.



Authentication on Mobile HTML Sites

- Keyboard imposes a lot of limitations in the way user accounts and passwords are defined.
- Smart phones have keyboards similar to traditional PC keyboards.
- Non PDA phones are limited to 0-9 keys with letters above numbers and special character support
- These create a challenge for creating strong passwords that require numbers, letters and special characters.
- Instead many banking and e-commerce mobile webs accept the user phone number as their account ID and a 4 digit password.

Encryption in Mobile Websites

- **Transport Layer Security** (**TLS**) and, **Secure Sockets Layer** (**SSL**), are security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral. It provide communication security over the Internet.
- The need for security, via SSL/TSL between the mobile device and its destination is equally important.
- This is available in the Wireless Application Protocol WAP 2.0

Application Attacks on Mobile HTML Sites

- Many traditional web application attacks will be also dangerous on mobile browser/devices, especially the ones we saw at the beginning of this session.

Cross-Site Scripting

- A session cookie allows users to be recognized within a website so any page changes or data selection is remembered from page to page. The most common example of this functionality is the shopping cart feature of any e-commerce site.
- Cross-Site Scripting(XSS) : Allows an attacker to steal a user’s session cookie which is then used to access a given web application as the victim and steal their information.
- Mobile HTML support of CSS and JavaScript make this kind of attacks possible.

HTTP Redirects and Phishing

- HTTP Redirects re-direct a victim to a page of the attacker’s choice.
- Phishing is an old attack that aims at harvesting and validating emails so as to sell them.
- Limited Viewing space on mobile devices makes them particularly vulnerable to these kinds of attacks because users do not have access or room to view the whole URL links and ascertain they are legitimate.

Mobile Geolocation

- Both Mobile and Desktop applications are increasingly making use of positional data to provide geolocation services.
- Geolocation on mobile devices used to be solely for emergency and law enforcement purposes.

Geolocation Methods

- Tower Triangulation: This method requires at least two towers and uses the relative power levels of radio signals between a cell phone and a cell tower of a known location.
- GPS: Using satellite signals instead of cell towers. It cannot be used indoors and initial GPS location may take several minutes to acquire
- 802.11: This method works by doing a survey of any nearby 802.11(Wi-Fi) wireless access point and submitting the data to a web service that returns the coordinates.

Geolocation Implementation

- Android: Permission to use the geolocation features is requested via program and is granted by the user. There are two permissions: ACCESS_COARSE_LOCATION(cell triangulation or Wi-Fi) or ACCESS_FINE_LOCATION(GPS)
- iPhone: Geolocation requires user approval every time an application using it, is launched. Uses the least precise measurement that meet the functionality requirement.
- Windows Mobile: Has no mechanism for a user to control geolocation access on app-by-app basis(all apps are allowed access if the service is enabled on the device).

Risks of Geolocation Services

Risks to the End User:

Geolocation services can violate a user privacy because his/her locations can be monitored at all times.

Positional data stored on remote servers introduces a new threat of data theft, can reveal information about the user history whereabouts.

Risks of Geolocation Services

Risks to Service Providers: By maintaining extended positional records on users, service providers:

- increase the risk of data breach,
- Increase the risk of legal and congressional subpoenas and
- possibly aid in illegal and criminal acts.

Geolocation Best Practices

- Only request the degree of accuracy that your app requires
- Discard data after use.
- Keep data anonymous: if data needs to be retained, ensure that it cannot be associated with other personal data.
- Users should be visually notified that their whereabouts are being recorded
- All software using geolocation data should have this functionality disabled until explicit confirmation from the user.
- Provide guarantees to your users about how their data is being used and or stored.

Bibliography

- Five common Web Applications vulnerabilities

http://www.symantec.com/connect/articles/five-common-web-application-vulnerabilities

- Cross-Site Scripting http://en.wikipedia.org/wiki/Cross-site_scripting
- Turning an HTTP proxy server into a Wireless internet gateway https://www.isoc.org/inet2000/cdproceedings/3b/3b_1.htm
- All about cookies http://www.allaboutcookies.org/cookies/session-cookies-used-for.html
- Mobile Application Security H. Dwivedi C. Clark, D. Thiel McGraw Hill Professional 2010